

CLAIMS

What is claimed is:

1. A method of reducing a time for a computer system to recover from a
degradation of performance in a hardware or a software in at least one first
5 node of said computer system, comprising:

monitoring a state of said at least one first node; and
based on said monitoring, transferring a state of said at least one first
node to a second node prior to said degradation in performance of said
hardware or said software of said at least one first node.

10 2. The method of claim 1, wherein said degradation of performance
comprises one of an outage and a failure.

3. The method of claim 1, further comprising:

predicting an outage of said hardware or said software based on
monitoring, and beginning said transferring based on said predicting.

15 4. The method of claim 1, further comprising:

proactively invoking a state migration functionality to reduce said
recovery time.

5. The method of claim 1, further comprising:

migrating a dynamic state to stable storage of said second node, said second node being accessible to a recovering agent, to reduce an amount of time required by said recovering agent.

5 6. The method of claim 1, wherein said computer system comprises a single node computer system.

7. The method of claim 1, wherein said computer system comprises a multi-node system.

10 8. The method of claim 1, wherein said second node selectively includes an application running corresponding to an application failing on said at least one first node.

9. The method of claim 1, further comprising:

15 connecting said at least one first node and said second node to a shared memory containing a stale state of the at least one first node and a redo log.

10. The method of claim 9, wherein said shared memory includes at least one of a shared storage medium, a shared storage disk and a shared network.

11. The method of claim 9, wherein a state transfer from said at least one first node to said second node occurs while the at least one first node is still operational.

12. The method of claim 9, further comprising:
providing a failure predictor on at least one of said at least one first
node and said second node, for commanding the at least one first node to start
5 an application if not already running, and commanding the second node to
begin reading a state of said at least one node and redo log from the shared
memory.

13. The method of claim 12, wherein said at least one node is commanded to
begin mirroring its dynamic state updates to the second node as they occur, in
10 an attempt to get the second node's state completely up to date.

14. The method of claim 1, further comprising:
scheduling a rejuvenation to avoid an unplanned failure.

15. The method of claim 1, further comprising:
predicting any of an application, hardware, and operating system of
15 said computer system as failing or undergoing a lack of performance.

16. The method of claim 1, further comprising:
bringing said second node's state into coincidence with the stale state
of the at least one first node undergoing a lack of performance, such that the
second node begins to mirror the at least one first node.

17. The method of claim 1, further comprising:

one of rejuvenating the at least one first node.

18. The method of claim 1, further comprising:

intentionally failing the at least one first node if said at least one first

5 node is undergoing a resource exhaustion failure; and

bringing the at least one first node back.

19. A method of reducing a lack of performance in a computer system having
at least one primary node and a secondary node, comprising:

10 determining whether a failure or lack of performance is imminent;

based on said determining, commanding a secondary node to start an
application if it is not already running, and to begin reading a state and redo
log from a memory coupled to said primary node and said secondary node;

commanding the secondary node to apply the redo log to its state;

15 commanding the primary node to begin mirroring its dynamic state
updates to the secondary node as they occur, such that the secondary node's
state is brought completely up to date with said primary node;

judging whether the primary node has failed; and

based on said judging, making the secondary node become the primary
20 node.

20. The method of claim 19, further comprising:

rebooting the primary node such that the primary node subsequently becomes the secondary node.

21. The method of claim 19, further comprising:

rejuvenating the primary node.

5 22. The method of claim 19, wherein no dedicated secondary node is provided for each said at least one primary node.

23. The method of claim 19, a one-to-many relationship exists between a number of said secondary node and said at least one primary node.

10 24. The method of claim 19, wherein said secondary node need not be located until it is judged that a potential chance for an outage or performance degradation occurs.

15 25. The method of claim 19, wherein said secondary node is provided for a plurality of primary nodes, and, when it is determined that one primary node is about to fail, the secondary node begins mirroring a state of the failing primary node.

26. A method of maintaining performance of a computer system, comprising:
monitoring a primary node of said computer system;
determining whether the primary node is failing or about to fail; and

migrating the state of the primary node to another node in said computer system,

wherein there is other than a one-to-one relationship between the another node and the primary node.

5 27. A method of reducing a degradation period of a Web hosting machine, comprising:

monitoring a performance of said Web hosting machine; and

transferring a state of said Web hosting machine to a second machine when a degradation of said performance occurs in said Web hosting machine.

10 28. A method of reducing a degradation of performance in a computer system having at least one primary node and a secondary node, comprising:

determining whether a degradation of performance of said primary node is imminent;

15 based on said determining, commanding said secondary node to start an application if it is not already running;

replicating, by said secondary node, a state of said primary node; and passing control to said secondary node from said primary node.

29. The method of claim 28, further comprising:

one of recovering and repairing the primary node.

20 30. The method of claim 28, further comprising:

YOR920010067US1

rejuvenating the primary node.

31. The method of claim 28, wherein said replicating comprises reading a stale state of said primary node and a redo log from a memory coupled to said primary node and said secondary node.

5 32. The method of claim 28, wherein the primary node is operational while said secondary node is replicating the state of the primary node.

33. The method of claim 28, wherein one said secondary node is provided for a plurality of ones of said primary node.

10 34. A method of reducing a degradation of performance in a computer system having a single node, comprising:

determining whether a degradation of performance of the node is imminent;

based on the determining, commanding the node to begin storing its state on a stable storage at a more frequent rate, to reduce a staleness of the state on the stable storage.

15

35. A system for reducing a time for a computer system to recover from a degradation of performance in a hardware or a software in at least one first node of said computer system, comprising:

a monitor for monitoring a state of said at least one first node; and

a transfer mechanism for, based on an output from said monitor, transferring a state of said at least one first node to a second node prior to said degradation in performance of said hardware or said software of said at least one first node.

5 36. A computer system, comprising:

at least one first node;

a second node;

a shared memory coupled to said first and second nodes;

a monitor for monitoring a state of said at least one first node; and

10 a transfer mechanism for, based on an output from said monitor,

transferring a state of said at least one first node to said second node prior to a degradation in performance of hardware or software of said at least one first node.

37. A system for reducing a degradation of performance in a computer system

15 having a single node and a stable storage, including:

a monitoring unit for monitoring whether a degradation of

performance of the node is imminent; and

a transfer mechanism for, based on an output from the determining

unit, commanding the node to begin storing its state on a stable storage at a

20 more frequent rate, to reduce a staleness of the state on the stable storage.

38. A signal bearing medium tangibly embodying a program of
machine-readable instructions executable by a digital processing apparatus to
perform a method of reducing a time for a computer system to recover from a
degradation of performance in a hardware or a software in at least one first
5 node of said computer system, said method comprising:

monitoring a state of said at least one first node; and

based on said monitoring, transferring a state of said at least one first
node to a second node prior to said degradation in performance of said
hardware or said software of said at least one first node.

10 39. A signal bearing medium tangibly embodying a program of
machine-readable instructions executable by a digital processing apparatus to
perform a method of reducing a degradation of performance in a computer
system having a single node, said method comprising:

15 determining whether a degradation of performance of the node is
imminent; and

based on the determining, commanding the node to begin storing its
state on a stable storage at a more frequent rate, to reduce a staleness of the
state on the stable storage.

20 40. A method of reducing a time for a computer system to recover from a
degradation of performance in a hardware or a software in a node of said
computer system, comprising:

monitoring a state of said node; and

based on said monitoring, transferring a state of said node to one of a stable storage and another node prior to said degradation in performance of said hardware or said software of said node.